AMENDMENTS TO THE SPECIFICATION

Please replace the paragraphs beginning on page 5, lines 1- page 6, line 23, with the following amended paragraphs:

Figure 1 is a schematic diagram of a software installation system 100 at an information handling system manufacturing site. In operation, an order 110 is placed to purchase a target ~~information handling~~ system 120. The target ~~information handling~~ system 120 to be manufactured contains a plurality of hardware and software components. For instance, target ~~information handling~~ system 120 might include a certain brand of hard drive, a particular type of monitor, a certain brand of processor and software. The software may include a particular version of an operating system along with all appropriate driver software and other application software along with appropriate software bug fixes. Before target ~~information handling~~ system 120 is shipped to the customer, the plurality of components are installed and tested. Such software installation and testing advantageously ensures a reliable, working information handling system which is ready to operate when received by a customer.

Because different families of information handling systems and different individual computer components require different software installation, it is necessary to determine which software to install on a target ~~information handling~~ system 120. A descriptor file 130 is provided by converting an order 110, which corresponds to a desired information handling system having desired components, into a computer readable format via ~~conversion~~ module conversion 132.

Component descriptors are computer readable descriptions of the components of target ~~information handling~~ system 120 which components are defined by the order 110. In an embodiment of the present invention, the component descriptors are included in a descriptor file called a system descriptor record which is a computer readable file containing a listing of the components, both hardware and software, to be installed onto target

~~information handling~~ system 120. Having read the plurality of component descriptors, database server 140 provides a plurality of software components corresponding to the component descriptors to file server 142 over network ~~connection~~ connections 144. Network connections 144 may be any network connection well-known in the art, such as a local area network, an intranet, or the internet. The information contained in database server 140 is often updated such that the database contains a new factory build environment. The software is then installed on the target ~~information handling~~ system 120. The software installation is controlled by a software installation management server, discussed in greater detail below, that is operable to control the installation of the operating system and other software packages specified by a customer.

Figure 2 is a generalized illustration of an information handling system, such as the target ~~information handling~~ system 120 illustrated in Figure 1. The information handling system includes a ~~processor~~ CPU 202, input/output (I/O) devices 204, such as a display, a keyboard, a mouse, and associated controllers, a hard disk drive 206, and other storage devices 208, such as a floppy disk and drive and other memory devices, and various other subsystems 210, all interconnected via one or more buses 212. The software that is installed according to the versioning methodology is installed onto hard disk drive 206. Alternately, the software may be installed onto any appropriate non-volatile memory. The non-volatile memory may also store the information relating to which factory build environment was used to install the software. Accessing this information enables a user to have additional systems corresponding to a particular factory build environment to be built.

Please replace the paragraphs on page 8, lines 6-20, with the following amended paragraphs:

Figure 4 is a conceptual illustration of the disassembled and decompressed program files A, B, …, G wherein each of the program files represents an independent program file

that can be used to reassemble a software application. These individual program files are stored in temporary directories and are then indexed and analyzed in a software dissemination server, discussed hereinbelow in Figure 6, to generate an index of program files representing all of the program files that are needed to generate a family of software applications, such as the software applications 302, ~~306~~ 304 and ~~308~~ 306 illustrated in Figure 3.

Figure 5A illustrates a composite program file image library 308 that contains all of the program files A, B, …, G that are necessary to create the original software applications, 302, 304, and ~~308~~ 306. The composite program file image library contains one copy of each of the unique program files; however, all redundant copies of the program files have been removed. The composite program file image library 308 can be stored in a program storage cache in a factory server, as discussed hereinbelow, to allow improved efficiency in the manufacture of information handling systems.

Please replace the paragraph on page 9, lines 3-15, with the following amended paragraph:

Figure 6 is an illustration of the components of the automated system for converting, optimizing and disseminating software in accordance with the present invention. Software applications 302, 304 and 306 are delivered to the system via a firewall 606 and are received by a software dissemination server 608. When a new software application is received, the software dissemination server 608 unpacks the application file into individual program files. The software dissemination server 608 scans the software applications for viruses and then transfers the packages to the repack and script regeneration server 612. The repack and script regeneration server 612 is operable to disassemble the software applications 302, 304, and 306 into individual program files. In this process, the repack and script regeneration server 612 disassembles and decompresses the software applications 302, 304, and 306 into the individual program files, such as the program files A, B, …, and G discussed hereinabove in

connection with Figures 3-6 and stores these individual program files in temporary directories.

Please replace the paragraphs beginning on page 9, line 23- page 10, line 20, with the following amended paragraphs:

The repack and script regeneration server 612 also constructs an index of the program files contained in each of the software applications 302, 304, 306. This index and the related program files are used to construct a composite program file image library that is transferred to the download server 616, which includes a software image cache 617. The composite program file image library contains one copy of each of the unique program files but, as discussed herein, all of the redundant copies of the program files have been removed. The program files, e.g., program files A, B, …, G, assembled into the composite program file image library 308, as discussed hereinabove, are stored in the software image cache 617 to decrease the time required to access these files, thereby improving the performance of the "build to order" system.

The repack and script regeneration server 612 generates top level factory scripts for each of the program files to produce "factory installable bits" corresponding to the various software applications. These "factory installable bits" are then transferred to the download server 616. Copies of the program file images are also transferred to an archive server 614. The repack and script regeneration server 612 then generates a signal authorizing the script and ~~installation~~ installer validation server 618 to generate appropriate commands to control downloading of software application to the target information handling system 120. The results of the installation are monitored by the script and installer validation server 618 and results are communicated to the software dissemination server 608 while the actual software images are downloaded by the download server 616 onto the hard drive or other storage media of the target information handling system 120.